On “A Novel Maximum Likelihood Decoding Algorithm for Orthogonal Space-Time Block Codes”

Ender Ayanoglu
Center for Pervasive Communications and Computing
Department of Electrical Engineering and Computer Science
University of California, Irvine

Abstract

The computational complexity of the Maximum Likelihood decoding algorithm in [1], [2] for orthogonal space-time block codes is smaller than specified.

I. Introduction

In [1],[2], the decoding of an Orthogonal Space-Time Block Code (OSTBC) in a Multi-Input Multi-Output (MIMO) system with N transmit and M receive antennas, and an interval of T symbols during which the channel is constant, is considered. The received signal is given by

$$Y = \mathcal{G}_N H + V \tag{1}$$

where $Y = [y_t^j]_{T\times M}$ is the received signal matrix of size $T \times M$ and whose entry y_t^j is the signal received at antenna j at time t, $t = 1, 2, \ldots, T$, $j = 1, 2 \ldots, M$; $V = [v_t^j]_{T\times M}$ is the noise matrix, and $\mathcal{G}_N = [g_t^j]_{T\times N}$ is the transmitted signal matrix whose entry g_t^i is the signal transmitted at antenna i at time t, $t = 1, 2, \ldots, N$. The matrix $H = [h_{i,j}]_{N\times M}$ is the channel coefficient matrix of size $N \times M$ whose entry $h_{i,j}$ is the channel coefficient from transmit antenna i to receive antenna j. The entries of the matrices H and V are independent, zero-mean, and circularly symmetric complex Gaussian random variables.

The real-valued representation of (1) is obtained by first arranging the matrices Y, H, and V, each in one column vector by stacking their columns one after the other as

$$\begin{bmatrix} y_1^1 \\ \vdots \\ y_T^M \end{bmatrix} = \check{\mathcal{G}}_N \begin{bmatrix} h_{1,1} \\ \vdots \\ h_{N,M} \end{bmatrix} + \begin{bmatrix} v_1^1 \\ \vdots \\ v_T^M \end{bmatrix} \tag{2}$$

where $\check{\mathcal{G}}_N = I_M \otimes \mathcal{G}_N$, with I_M denoting the identity matrix of size M and $\otimes$ denoting the Kronecker matrix multiplication, and then decomposing the MT-dimensional complex problem defined by (2) to a $2MT$-dimensional real-valued problem by applying the real-valued lattice representation defined in [3] to obtain

$$\check{y} = \check{H}x + \check{v} \tag{3}$$

or equivalently

$$\begin{bmatrix} \text{Re}(y_1^1) \\ \text{Im}(y_1^1) \\ \vdots \\ \text{Re}(y_T^M) \\ \text{Im}(y_T^M) \end{bmatrix} = \check{H} \begin{bmatrix} \text{Re}(s_1) \\ \text{Im}(s_1) \\ \vdots \\ \text{Re}(s_K) \\ \text{Im}(s_K) \end{bmatrix} + \begin{bmatrix} \text{Re}(v_1^1) \\ \text{Im}(v_1^1) \\ \vdots \\ \text{Re}(v_T^M) \\ \text{Im}(v_T^M) \end{bmatrix}. \tag{4}$$

The real-valued fading coefficients of $\check{H}$ are defined using the complex fading coefficients $h_{i,j}$ from transmit antenna i to receive antenna j as $h_{2l-1+2(j-1)N} = \text{Re}(h_{l,j})$ and $h_{2l+2(j-1)N} = \text{Im}(h_{l,j})$ for

$l = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$. Since $\mathcal{G}_N$ is an orthogonal matrix and due to the real-valued representation of the system using (4), it can be observed that

- All columns of $\check{H} = [\check{h}_1 \check{h}_2 \ldots \check{h}_{2K}]$ where $\check{h}_i$ is the ith column of $\check{H}$, are orthogonal to each other, or equivalently
$$\check{h}_i^T \check{h}_j = 0 \qquad i, j = 1, 2, \ldots, K, i \neq j \tag{5}$$
- The inner product of every column in $\check{H}$ with itself is equal to a constant, i.e.,
$$\check{h}_i^T \check{h}_i = \check{h}_j^T \check{h}_j \qquad i, j = 1, 2, \ldots, K. \tag{6}$$

II. Decoding

Let
$$\sigma = \check{h}_1^T \check{h}_1. \tag{7}$$
Note $\sigma = \check{h}_i^T \check{h}_i$, $i = 1, 2, \ldots, 2K$. Due to the orthogonality property in (5)-(6), we have
$$\check{H}^T \check{H} = \sigma I_{2K}. \tag{8}$$
Let's represent (4) as
$$\check{y} = \check{H} x + \check{v}. \tag{9}$$
By multiplying this expression by $\check{H}^T$ on the left, we have
$$\begin{aligned} \bar{y} &= \check{H}^T \check{y} && (10) \\ &= \sigma x + \bar{v} && (11) \end{aligned}$$
where $\bar{v}$ is zero-mean, and due to (8) has independent and identically distributed Gaussian members. The Maximum Likelihood solution is found by minimizing
$$\left\| \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_{2K} \end{bmatrix} - \sigma \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_{2K} \end{bmatrix} \right\|_2^2 \tag{12}$$
over all combinations of $x \in \Omega^{2K}$. This can be further simplified as
$$\hat{x}_i = \arg \min_{x_i \in \Omega} |\bar{y}_i - \sigma x_i|^2 \tag{13}$$
for $i = 1, 2, \ldots, 2K$. Then, the decoded message is
$$\hat{x} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{2K})^T. \tag{14}$$

III. Computational Complexity

The decoding operation consists of the multiplication $\check{H}^T \check{y}$, calculation of $\sigma = \check{h}_1^T \check{h}_1$, the multiplications σx_i, and performing (13). With a slight change, we will consider the calculation of σ^{-1} and the multiplications
$$z_i = \sigma^{-1} \bar{y}_i \qquad i = 1, 2, \ldots, 2K. \tag{15}$$
Then
$$\hat{x}_i = \arg \min_{x_i \in \Omega} |z_i - x_i|^2 \tag{16}$$
for $i = 1, 2, \ldots, 2K$, which is a standard quantization operation in conventional Quadrature Amplitude Modulation. We will compute the decoding complexity up to this quantization operation. Note $\check{H}$ is a $2MT \times 2K$ matrix, $\check{h}_1$ is a $2MT$-dimensional vector, and we will assume the complexity of real division

as equivalent to 4 real multiplications as in [1],[2]. The multiplication $\check{H}^T\check{y}$ takes $2MT \cdot 2K$, calculation of σ takes $2MT$, its inverse takes 4, and $\sigma^{-1}\bar{y}$ takes $2K$ real multiplications. Similarly, the multiplication $\check{H}^T\check{y}$ takes $2K \cdot (2MT - 1)$, and calculation of σ takes $2MT - 1$ real additions. Letting R_M and R_A be the number of real multiplications and the number of real additions, the complexity of decoding the transmitted complex signal $(s_1, s_2, \ldots, s_K)$ with the technique described in (7), (10), and (15) is

$$\mathcal{C}_{\text{PR}} = (4KMT + 2MT + 2K + 4)R_M, (4KMT + 2MT - 2K - 1)R_A \tag{17}$$

which is smaller than the complexity specified in [1],[2] and does not depend on the constellation size L. However, as will be seen in the examples, the matrix $\check{H}$ can include values identical to 0, or multiplications by a scalar, which result in deviations from (17). Also, in (56), we will provide a slightly smaller figure for this complexity. In what follows, we will calculate the exact complexity values for four examples.

Due to the orthogonality property (8) of $\check{H}$, the QR decomposition of $\check{H}$ is

$$Q = \frac{1}{\sqrt{\sigma}}\check{H} \qquad R = \sqrt{\sigma} I_{2K} \tag{18}$$

and therefore does not need to be computed explicitly. The procedure described above is equivalent and has lower computational complexity.

IV. Comparison with a Conventional Technique

We will now compare the technique in (7), (10), and (15) with one from the literature. In [4], it has been shown that

$$\|Y - \mathcal{G}_N H\|^2 = \|H\|^2 \sum_{k=1}^{K} |s_k - \hat{s}_k|^2 + \text{constant}, \tag{19}$$

where

$$\hat{s}_k = \frac{1}{\|H\|^2}[\text{Re}\{\text{Tr}(H^H A_k^H Y)\} - \hat{\imath} \cdot \text{Im}\{\text{Tr}(H^H B_k^H Y)\}] \tag{20}$$

and where A_k and B_k are the matrices in the linear representation of $\mathcal{G}_N$ in terms of $\bar{s}_k = \text{Re}[s_k]$ and $\tilde{s}_k = \text{Im}[s_k]$ for $k = 1, 2, \ldots, K$ as [4]

$$\mathcal{G}_N = \sum_{k=1}^{K} \bar{s}_k A_k + \hat{\imath}\tilde{s}_k B_k = \sum_{k=1}^{K} s_k \check{A}_k + s_k^* \check{B}_k, \tag{21}$$

$\hat{\imath} = \sqrt{-1}$, $A_k = \check{A}_k + \check{B}_k$, and $B_k = \check{A}_k - \check{B}_k$. Once $\{\hat{s}_k\}_{k=1}^{K}$ are calculated, the decoding problem can be solved by

$$\min_{s_k \in \Omega^2} |s_k - \hat{s}_k|^2 \tag{22}$$

once for each $k = 1, 2, \ldots, K$. Similarly to (16), this is a standard quantization problem in Quadrature Amplitude Modulation and we will calculate the computational complexity of this approach up to this point.

We will carry out the computational complexity analysis of the technique in (7), (10), and (15) against the complexity of the technique in (20) for four examples, including those in [1], [2].

Example 1: Consider the Alamouti OSTBC with $N = K = T = 2$ and $M = 1$ where

$$\mathcal{G}_2 = \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix}. \tag{23}$$

The received signal is given by

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix} \begin{bmatrix} h_{1,1} \\ h_{2,1} \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{24}$$

Representing (24) in the real domain, we have

$$\begin{bmatrix} \mathrm{Re}(y_1) \\ \mathrm{Im}(y_1) \\ \mathrm{Re}(y_2) \\ \mathrm{Im}(y_2) \end{bmatrix} = \check{H} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \mathrm{Re}(v_1) \\ \mathrm{Im}(v_1) \\ \mathrm{Re}(v_2) \\ \mathrm{Im}(v_2) \end{bmatrix} \tag{25}$$

where $x_1 = \mathrm{Re}(s_1)$, $x_2 = \mathrm{Im}(s_1)$, $x_3 = \mathrm{Re}(s_2)$, $x_4 = \mathrm{Im}(s_2)$ and

$$\check{H} = \begin{bmatrix} h_1 & -h_2 & h_3 & -h_4 \\ h_2 & h_1 & h_4 & h_3 \\ h_3 & h_4 & -h_1 & -h_2 \\ h_4 & -h_3 & -h_2 & h_1 \end{bmatrix}. \tag{26}$$

Note that the matrix $\check{H}$ is orthogonal and all of its columns have the same squared norm. One needs 16 real multiplications to calculate $\bar{y} = \check{H}^T \check{y}$, 4 real multiplications to calculate $\sigma = \check{h}_1^T \check{h}_1$, 4 real multiplications to calculate σ^{-1}, and 4 real multiplications to calculate $\sigma^{-1}\bar{y}$. There are $3 \cdot 4 = 12$ real additions to calculate $\check{H}^T \check{y}$ and 3 real additions to calculate σ. As a result, with this approach, decoding takes a total of 28 real multiplications and 15 real additions.

For the method in (20) above, the products $H^H A_1^H$, $H^H B_1^H$, $H^H A_2^H$, $H^H B_2^H$ are

$$\begin{array}{cc} H^H A_1^H = [\ h_{1,1}^* \quad h_{2,1}^*\ ] & H^H B_1^H = [\ h_{1,1}^* \quad -h_{2,1}^*\ ] \\ H^H A_2^H = [\ h_{2,1}^* \quad -h_{1,1}^*\ ] & H^H B_2^H = [\ h_{2,1}^* \quad h_{1,1}^*\ ] \end{array} \tag{27}$$

which will be multiplied by $Y = (y_1, y_2)^T$ where $h_{1,1}$, $h_{2,1}$, y_1, y_2 are all complex. It can be observed from (20) and (27) that one needs all products $h_{i,1}^* y_j$, $i, j = 1, 2$. Therefore, one needs 4 complex or 16 real multiplications. The calculation of $\|H\|^2$ takes 4, its reciprocal $1/\|H\|^2$ 4, and the multiplication of $1/\|H\|^2$ with $\mathrm{Re}\{\mathrm{Tr}[H^H A_k^H Y]\}$ and $\mathrm{Im}\{\mathrm{Tr}[H^H B_k^H Y]\}$ for $k = 1, 2$ another 4 real multiplications. It can be calculated that each of $\mathrm{Re}\{\mathrm{Tr}[H^H A_k^H Y]\}$ and $\mathrm{Im}\{\mathrm{Tr}[H^H B_k^H Y]\}$ has 3 distinct real additions for $k = 1, 2$, which means there are a total of 12 real additions for this operation. Calculation of $\|H\|^2$ takes 3 real additions. As a result, this approach employs 28 real multiplications and 15 real additions to decode.

Note, in this case, the complexity figures in (17) are 28 real multiplications and 15 real additions, which hold exactly.

Example 2: Consider the OSTBC with $M = 2$, $N = 3$, $T = 8$, and $K = 4$ given by [5]

$$\mathcal{G}_3 = \begin{bmatrix} s_1 & -s_2 & -s_3 & -s_4 & s_1^* & -s_2^* & -s_3^* & -s_4^* \\ s_2 & s_1 & s_4 & -s_3 & s_2^* & s_1^* & s_4^* & -s_3^* \\ s_3 & -s_4 & s_1 & s_2 & s_3^* & -s_4^* & s_1^* & s_2^* \end{bmatrix}^T. \tag{28}$$

The received signal can be written as

$$\begin{bmatrix} y_1^1 & y_1^2 \\ \vdots & \vdots \\ y_8^1 & y_8^2 \end{bmatrix} = \mathcal{G}_3 \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \\ h_{3,1} & h_{3,2} \end{bmatrix} + \begin{bmatrix} v_1^1 & v_1^2 \\ \vdots & \vdots \\ v_8^1 & v_8^2 \end{bmatrix}. \tag{29}$$

In [2], it has been shown that the 32×8 real-valued channel matrix $\check{H}$ is

$$\check{H} = \begin{bmatrix} h_1 & -h_2 & h_3 & -h_4 & h_5 & -h_6 & 0 & 0 \\ h_2 & h_1 & h_4 & h_3 & h_6 & h_5 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_7 & -h_8 & h_9 & -h_{10} & h_{11} & -h_{12} & 0 & 0 \\ h_8 & h_7 & h_{10} & h_9 & h_{12} & h_{11} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & h_{11} & h_{12} & -h_9 & -h_{10} & -h_7 & -h_8 \\ 0 & 0 & h_{12} & -h_{11} & -h_{10} & h_9 & -h_8 & h_7 \end{bmatrix} \tag{30}$$

where h_i, $i = 1, 2, \ldots, 11$ and h_j, $j = 2, 4, \ldots, 12$ are the real and imaginary parts, respectively, of $h_{1,1}$, $h_{2,1}$, $h_{3,1}$, $h_{1,2}$, $h_{2,2}$, $h_{3,2}$. The matrix $\check{H}^T$ is 8×32 where each row has 8 zeros, while each of the remaining 24 symbols has one of $h_1, h_2, \ldots, h_{12}$, repeated twice. Let's first ignore the repetition of h_i in a row. Then, the calculation of $\check{H}^T \check{y}$ takes $8 \cdot 24 = 192$ real multiplications. The calculation of $\sigma = \hat{h}_1^T \hat{h}_1 = 2 \sum_{k=1}^{12} h_i^2$ takes $12 + 1 = 13$ real multiplications, In addition, one needs 4 real multiplications to calculate σ^{-1}, and 8 real multiplications to calculate $\sigma^{-1} \bar{y}$. To calculate $\check{H}^T \check{y}$, one needs $8 \cdot 23 = 184$ real additions, and to calculate σ, one needs 11 real additions. As a result, with this approach, one needs a total of 217 real multiplications and 195 real additions to decode.

For the method in (20) above, the products $H^H A_1^H$ and $H^H B_1^H$ are

$$\begin{aligned} H^H A_1^H &= \begin{bmatrix} h_{1,1}^* & h_{2,1}^* & h_{3,1}^* & 0 & h_{1,1}^* & h_{2,1}^* & h_{3,1}^* & 0 \\ h_{1,2}^* & h_{2,2}^* & h_{3,2}^* & 0 & h_{1,2}^* & h_{2,2}^* & h_{3,2}^* & 0 \end{bmatrix} \\ H^H B_1^H &= \begin{bmatrix} h_{1,1}^* & h_{2,1}^* & h_{3,1}^* & 0 & -h_{1,1}^* & -h_{2,1}^* & -h_{3,1}^* & 0 \\ h_{1,2}^* & h_{2,2}^* & h_{3,2}^* & 0 & -h_{1,2}^* & -h_{2,2}^* & -h_{3,2}^* & 0 \end{bmatrix} \end{aligned} \tag{31}$$

Other $H^H A_k^H$ and $H^H B_k^H$ have similar structures, with zero columns located elsewhere, same location in $H^H A_k^H$ and $H^H B_k^H$, $k = 2, 3, 4$. Nonzero columns of $H^H A_k^H$ and $H^H B_k^H$ are the shuffled versions of the columns of $H^H A_1^H$ and $H^H B_1^H$, with the same shuffling for $H^H A_k^H$ and $H^H B_k^H$, possibly with sign changes. As a result, the first four columns of $H^H A_k^H$ and $H^H B_k^H$ are the same, the first and second four columns of $H^H A_k^H$ are the same, while the first and second four columns of $H^H B_k^H$ are negatives of each other, $k = 1, 2, 3, 4$. For this $\mathcal{G}_N$, one has

$$\mathcal{G}_N^H \mathcal{G}_N = 2 \left(\sum_{k=1}^{K} |s_k|^2 \right) I \tag{32}$$

which makes it necessary to replace $\|H\|^2$ with $2\|H\|^2$ in (20) above. The vector Y is given as

$$Y = \begin{bmatrix} y_1^1 & y_1^2 \\ \vdots & \vdots \\ y_8^1 & y_8^2 \end{bmatrix}. \tag{33}$$

The complex multiplications in calculating $\mathrm{Re}\{\mathrm{Tr}[H^H A_1^H Y]\}$ can be used to calculate $\mathrm{Im}\{\mathrm{Tr}[H^H B_1^H Y]\}$ due to sign changes and the calculation of real and imaginary parts. Ignoring the repetition of $h_{i,j}^*$, there are 12 different complex numbers in $H^H A_1^H$ and due to the trace operation, they will be multiplied with 12 complex numbers from Y. As a result, to calculate $\mathrm{Tr}[H^H A_k^H Y]$ (equivalently $\mathrm{Tr}[H^H B_k^H Y]$) one needs 12 complex or 48 real multiplications for one k. To calculate the numerators of s_k, for all $k = 1, 2, 3, 4$, one needs 192 real multiplications. To calculate $2\|H\|^2$ in the denominator, one needs 13 real multiplications. To calculate its inverse, one needs 4 real multiplications. Finally, to complete the calculation of s_k for $k = 1, 2, 3, 4$ by multiplying the numerators of their real and imaginary parts by $1/(2\|H\|^2)$, one needs 8 real multiplications. To calculate each $\mathrm{Re}\{\mathrm{Tr}[H^H A_k^H Y]\}$ or $\mathrm{Im}\{\mathrm{Tr}[H^H B_k^H Y]\}$ for $k = 1, 2, 3, 4$, one needs $12 + 11 = 23$ real additions. To calculate $\|H\|^2$, one needs 11 additions. As a result, with this approach, one needs 217 real multiplications and 195 real additions to decode, same number as in the approach specified by (7), (10), and (15).

For this example, (17) specifies 300 real multiplications and 279 real additions. The reduction is due to the elements with zero values in $\check{H}$.

It is important to make the observation that the repeated values of h_i in the columns of $\check{H}$, or equivalently $h_{m,n}^*$ in the rows of $H^H A_k^H$ or $H^H B_k^H$, have a substantial impact on complexity. We will carry out the rest of this discussion only for the approach in (7), (10), and (15), the one in (20) is essentially the same. Due to the repetition of h_i, by grouping the two values of $\check{y}_j$ that it multiplies, it takes $8 \cdot 12 = 96$ real multiplications to compute $\check{H}^T \check{y}$, not $8 \cdot 24 = 192$. The summations for each row of $\check{H}^T \check{y}$ will now be done in two steps, first 12 pairs of additions per each h_i, and then after multiplication by h_i, addition of 12 real numbers. This takes $12 + 11 = 23$ real additions, with no change from the way the calculation was made without grouping. With this change, the complexity of decoding becomes 121 real multiplications and 195 real additions, a huge reduction from 300 real multiplications and 279 real additions.

Example 3: We will now consider the code $\mathcal{G}_4$ from [5]. The parameters for this code are $N = K = 4$, $M = 1$, and $T = 8$. It is given as

$$\mathcal{G}_4 = \begin{bmatrix} s_1 & -s_2 & -s_3 & -s_4 & s_1^* & -s_2^* & -s_3^* & -s_4^* \\ s_2 & s_1 & s_4 & -s_3 & s_2^* & s_1^* & s_4^* & -s_3^* \\ s_3 & -s_4 & s_1 & s_2 & s_3^* & -s_4^* & s_1^* & s_2^* \\ s_4 & s_3 & -s_2 & s_1 & s_4^* & s_3^* & -s_2^* & s_1^* \end{bmatrix}^T . \tag{34}$$

Similarly to $\mathcal{G}_3$ of Example 2, this code has the property that $\mathcal{G}_4^H \mathcal{G}_4 = 2(\sum_{k=1}^K |s_k|^2) I$. As a result, $\|H\|^2$ in the denominator in (20) should be replaced with $2\|H\|^2$. The $\check{H}$ matrix is 16×8 and can be calculated as

$$\check{H} = \begin{bmatrix} h_1 & -h_2 & h_3 & -h_4 & h_5 & -h_6 & h_7 & h_8 \\ h_2 & h_1 & h_4 & h_3 & h_6 & h_5 & h_8 & h_7 \\ h_3 & -h_4 & -h_1 & h_2 & h_7 & -h_8 & -h_5 & h_6 \\ h_4 & h_3 & -h_2 & -h_1 & h_8 & h_7 & -h_6 & -h_5 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_5 & h_6 & -h_7 & h_8 & -h_1 & -h_2 & h_3 & h_4 \\ h_6 & -h_5 & -h_8 & h_7 & -h_2 & h_1 & h_4 & -h_3 \end{bmatrix} . \tag{35}$$

This matrix consists entirely of nonzero entries. Each entry in a column equals $\pm h_i$ for some $i \in \{1, 2, \ldots, 8\}$, every h_i appearing twice in a column. Ignoring this repetition for now, calculation of $\check{H}^T \check{y}$ takes $8 \cdot 16 = 128$ real multiplications. Calculation of σ takes 9 real multiplications, its inverse 4 real multiplications, and the calculation of $\sigma^{-1} \bar{y}$ takes 8 real multiplications. Calculation of $\check{H}^T \check{y}$ takes $8 \cdot 15 = 120$ real additions, and calculation of σ takes 7 real additions. As a result, with this approach, to decode, one needs 149 real multiplications and 127 real additions.

For this code, for the method in (20), the matrices $H^H A_k^H$ and $H^H B_k^H$ $k = 1, 2, 3, 4$ are as follows.

$$\begin{aligned}
H^H A_1^H &= [\ h_{1,1}^* \ \ h_{2,1}^* \ \ h_{3,1}^* \ \ h_{4,1}^* \ \ h_{1,1}^* \ \ h_{2,1}^* \ \ h_{3,1}^* \ \ h_{4,1}^* \ ] \\
H^H A_2^H &= [\ h_{2,1}^* \ \ -h_{1,1}^* \ \ -h_{4,1}^* \ \ h_{3,1}^* \ \ h_{2,1}^* \ \ -h_{1,1}^* \ \ -h_{4,1}^* \ \ h_{3,1}^* \ ] \\
H^H A_3^H &= [\ h_{3,1}^* \ \ h_{4,1}^* \ \ -h_{1,1}^* \ \ -h_{2,1}^* \ \ h_{3,1}^* \ \ h_{4,1}^* \ \ -h_{1,1}^* \ \ -h_{2,1}^* \ ] \\
H^H A_4^H &= [\ h_{4,1}^* \ \ -h_{3,1}^* \ \ h_{2,1}^* \ \ -h_{1,1}^* \ \ h_{4,1}^* \ \ -h_{3,1}^* \ \ h_{2,1}^* \ \ -h_{1,1}^* \ ] \\
H^H B_1^H &= [\ h_{1,1}^* \ \ h_{2,1}^* \ \ h_{3,1}^* \ \ h_{4,1}^* \ \ -h_{1,1}^* \ \ -h_{2,1}^* \ \ -h_{3,1}^* \ \ -h_{4,1}^* \ ] \\
H^H B_2^H &= [\ h_{2,1}^* \ \ -h_{1,1}^* \ \ -h_{4,1}^* \ \ h_{3,1}^* \ \ -h_{2,1}^* \ \ h_{1,1}^* \ \ h_{4,1}^* \ \ -h_{3,1}^* \ ] \\
H^H B_3^H &= [\ h_{3,1}^* \ \ h_{4,1}^* \ \ -h_{1,1}^* \ \ -h_{2,1}^* \ \ -h_{3,1}^* \ \ -h_{4,1}^* \ \ h_{1,1}^* \ \ h_{2,1}^* \ ] \\
H^H B_4^H &= [\ h_{4,1}^* \ \ -h_{3,1}^* \ \ h_{2,1}^* \ \ -h_{1,1}^* \ \ -h_{4,1}^* \ \ h_{3,1}^* \ \ -h_{2,1}^* \ \ h_{1,1}^* \ ]
\end{aligned} \tag{36}$$

From this set we conclude that the complex multiplications between $H^H A_k^H Y$ and $H^H B_k H Y$ can be shared for a given $k = 1, 2, 3, 4$. The number of real multiplications to calculate $H^H A_k^H Y$ for all $k = 1, 2, 3, 4$ is $4 \cdot 8 \cdot 4 = 128$. The number of real multiplications to calculate $2\|H\|^2$ is $6 + 1 = 7$, and to calculate its inverse takes 4 real multiplications. Finally, the number of real multiplications to complete the calculation of s_k for all $k = 1, 2, 3, 4$ is 8. In order to calculate $H^H A_k^H Y$ or $H^H B_k^H Y$, one needs 8 real additions to perform each complex multiplication and 7 real additions to calculate the sum. As a result, calculation of $\mathrm{Re}\{\mathrm{Tr}[H^H A_k^H Y]\}$ and $\mathrm{Im}\{\mathrm{Tr}[H^H B_k^H Y]\}$ for all $k = 1, 2, 3, 4$ takes 8×15 real additions. Calculation of $\|H\|^2$ takes $6 + 1 = 7$ real additions. Therefore, with this approach the number of real multiplications and additions to decode are 149 and 127, respectively, same as the numbers needed for the approach in (7), (10), and (15).

For this example, equation (17) specifies 156 real multiplications and 135 real additions. The reduction is due to the fact that one row of $\check{H}^T$ has each h_i appearing twice. This reduces the number of multiplications and summations to calculate σ by about a factor of 2.

However, because each h_i appears twice in every row of $\check{H}^T$, the number of multiplications can actually be reduced substantially, as we discussed in Example 2. As discussed in Example 2, we can reduce the number of multiplications to calculate $\check{H}^T\check{y}$ by grouping the two multipliers of each h_i by summing them prior to multiplication by h_i, $i = 1, 2, \ldots, 8$. As seen in Example 2, this does not alter the number of real additions. With this simple change, the number of real multiplications to decode becomes 85 and the number of real additions to decode remains at 127.

Example 4: It is instructive to consider the code $\mathcal{H}_3$ given in [5] with $N = 3$, $K = 3$, $T = 4$ which we will consider for $M = 1$ where

$$\mathcal{H}_3 = \begin{bmatrix} s_1 & s_2 & s_3/\sqrt{2} \\ -s_2^* & s_1^* & s_3/\sqrt{2} \\ s_3^*/\sqrt{2} & s_3^*/\sqrt{2} & (-s_1 - s_1^* + s_2 - s_2^*)/2 \\ s_3^*/\sqrt{2} & -s_3^*/\sqrt{2} & (s_2 + s_2^* + s_1 - s_1^*)/2 \end{bmatrix}. \tag{37}$$

For this code, $\mathcal{H}_3^H \mathcal{H}_3 = (\sum_{k=1}^3 |s_k|^2) I$ is satisfied. In this case, the matrix $\check{H}$ can be calculated as

$$\check{H} = \begin{bmatrix} h_1 & -h_2 & h_3 & -h_4 & h_5/\sqrt{2} & -h_6/\sqrt{2} \\ h_2 & h_1 & h_4 & h_3 & h_6/\sqrt{2} & h_5/\sqrt{2} \\ h_3 & h_4 & -h_1 & -h_2 & h_5/\sqrt{2} & -h_6/\sqrt{2} \\ h_4 & -h_3 & -h_2 & h_1 & h_6/\sqrt{2} & h_5/\sqrt{2} \\ -h_5 & 0 & 0 & -h_6 & (h_1 + h_3)/\sqrt{2} & (h_2 + h_4)/\sqrt{2} \\ -h_6 & 0 & 0 & h_5 & (h_2 + h_4)/\sqrt{2} & -(h_1 + h_3)/\sqrt{2} \\ 0 & h_6 & h_5 & 0 & (h_1 - h_3)/\sqrt{2} & (h_2 - h_4)/\sqrt{2} \\ 0 & -h_5 & h_6 & 0 & (h_2 - h_4)/\sqrt{2} & (-h_1 + h_3)/\sqrt{2} \end{bmatrix}. \tag{38}$$

It can be verified that every column $\check{h}_i$ of $\check{H}$ has the property that $\check{h}_i^T \check{h}_i = \sigma = \|H\|^2 = \sum_{k=1}^6 h_k^6$ for $i = 1, 2, \ldots, 6$. In this case, the number of real multiplications to calculate $\check{H}^T\check{y}$ requires more caution than the previous examples. For the first four rows of $\check{H}^T$, this number is 6 real multiplications per row. For the last two rows, due to combining, e.g., h_1 and h_3 in $(h_1 + h_3)/\sqrt{2}$ in the fifth element of $\check{h}_5$, and the commonality of h_5 and h_6 for the first and third, and second and fourth, respectively, elements of $\check{h}_5$, and one single multiplier $1/\sqrt{2}$ for the whole column, the number of real multiplications needed is 7. As a result, calculation of $\check{H}^T\check{y}$ takes 38 real multiplications. Calculation of σ takes 6 real multiplications. One needs 4 real multiplications to calculate σ^{-1}, and 6 real multiplications to calculate $\sigma^{-1}\bar{y}$. First four rows of $\check{H}^T\check{y}$ require 5 real additions each. Last two rows of $\check{H}^T\check{y}$ require $4 + 7 = 11$ real additions each. This is a total of 42 real additions to calculate $\check{H}^T\check{y}$. Calculation of σ requires 5 real additions. Overall, with this approach one needs 54 real multiplications and 47 real additions to decode.

For this code, for the method in (20) above, the matrices $H^H A_k^H$ and $H^H B_k^H$, $k = 1, 2, 3$ are as follows.

$$\begin{aligned} H^H A_1^H &= [\ h_{1,1}^* \quad h_{2,1}^* \quad -h_{3,1}^* \quad 0\ ] \\ H^H A_2^H &= [\ h_{2,1}^* \quad -h_{1,1}^* \quad 0 \quad h_{3,1}^*\ ] \\ H^H A_3^H &= \frac{1}{\sqrt{2}}[\ h_{3,1}^* \quad h_{3,1}^* \quad h_{1,1}^* + h_{2,1}^* \quad h_{1,1}^* - h_{2,1}^*\ ] \\ H^H B_1^H &= [\ h_{1,1}^* \quad -h_{2,1}^* \quad 0 \quad h_{3,1}^*\ ] \\ H^H B_2^H &= [\ h_{2,1}^* \quad h_{1,1}^* \quad h_{3,1}^* \quad 0\ ] \\ H^H B_3^H &= \frac{1}{\sqrt{2}}[\ h_{3,1}^* \quad h_{3,1}^* \quad -h_{1,1}^* - h_{2,1}^* \quad -h_{1,1}^* + h_{2,1}^*\ ] \end{aligned} \tag{39}$$

Before discussing the complexity of the approach in (20), we would like to make an observation. A careful examination shows that the complex multiplications between $H^H A_k^H Y$ for $k = 1, 2, 3$ and $H^H B_j^H Y$ for $j = 1, 2, 3$ can be shared in the method outlined in (20). In this case, since h_3^* in the first and second element of $H^H A_3^H$ can be shared, there are 9 complex multiplications needed for the calculation of $H^H A_k^H Y$ for $k = 1, 2, 3$. The real values of those will be used in calculating the real parts of s_k, $k = 1, 2, 3$, and the imaginary parts in calculating the imaginary parts of s_k, $k = 1, 2, 3$, albeit in possibly different signs or locations. This requires a careful implementation where the needed complex multiplications are calculated, stored, and their real and imaginary parts carefully distributed in the most judicious manner. The 9 complex multiplications correspond to 36 real multiplications, and there are 2 more real multiplications, by $1/\sqrt{2}$ for the real and imaginary parts of s_3. As in the previous method, 6 real multiplications are needed to calculate $\|H\|^2$, 4 real multiplications to calculate $1/\|H\|^2$, and 6 real multiplications to complete the calculation of s_k, $k = 1, 2, 3$. The calculation of $\mathrm{Re}\{\mathrm{Tr}[H^H A_k^H Y]\}$ and $\mathrm{Im}\{\mathrm{Tr}[H^H B_k^H Y]\}$ for all $k = 1, 2, 3$ takes $4 \cdot 5 + 2 \cdot (6 + 5) = 42$ real additions, and the calculation of $\|H\|^2$ takes 5 more real additions. This approach results in a total of 54 real multiplications and 47 real additions to decode, as in the technique in (7), (10), and (15).

For this example, (17) specifies 66 real multiplications and 49 real additions. The reduction is due to the presence of the zero entries in $\check{H}$. On the other hand, the presence of the factor $1/\sqrt{2}$ in the last two rows of $\check{H}^T$ adds two real multiplications to the total number of real multiplications.

Before concluding this example, we would like to display the matrices A_3 and B_3 for this code.

$$A_3 = \begin{bmatrix} 0 & 0 & 1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \end{bmatrix} \qquad B_3 = \begin{bmatrix} 0 & 0 & 1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} \\ -1/\sqrt{2} & -1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \end{bmatrix} \tag{40}$$

In all other A_k and B_k matrices in the four examples studied, the entries were ±1. Furthermore, in all other A_k and B_k matrices in the four examples, there was at most one nonzero value in a row. In both A_3 and B_3 above, the entries are irrational numbers and two rows have two nonzero entries.

From the examples above, by studying the operations of the two techniques in detail, it can actually be seen that, not only is the computational complexity of the technique in (7), (11), and (15) is the same as the technique in (20), but also they actually perform equivalent operations.

V. Orthogonality of $\check{H}$ and Computational Complexity Revisited

We have seen in the examples that when $\mathcal{G}_N^H \mathcal{G}_N = c(\sum_{k=1}^K |s_k|^2)I$ where $c = 1, 2$, then $\sigma = c\|H\|^2$. We will now show this holds in general. Based on that result, we will then reduce the computational complexity estimate in (17).

Let

$$z = \mathrm{vec}(Y) = \begin{bmatrix} y_1^1 \\ \vdots \\ y_T^M \end{bmatrix}. \tag{41}$$

Form two vectors, $\bar{s}$ and $\tilde{s}$, consisting of real and imaginary parts of s_k, and form a vector s' that is the concatenation of $\bar{s}$ and $\tilde{s}$:

$$\bar{s} = (\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_K)^T, \quad \tilde{s} = (\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_K)^T, \quad s' = (\bar{s}, \tilde{s})^T. \tag{42}$$

By rearranging the right hand side of (2), we can write

$$z = Fs' + e = F_a\bar{s} + F_b\tilde{s} + e \tag{43}$$

where $F = [F_a\ F_b]$ is an $MT \times 2K$, and F_a and F_b are $MT \times K$ complex matrices whose entries consist of (linear combinations of) channel coefficients $h_{i,j}$, and e is the corresponding complex Gaussian noise vector. In [4], it was shown that when $\mathcal{G}_N^H\mathcal{G}_N = (\sum_{k=1}^K |s_k|^2)I$, then $\mathrm{Re}[F^HF] = \|H\|^2 I$. It is straightforward to extend this result so that when $\mathcal{G}_N^H\mathcal{G}_N = c(\sum_{k=1}^K |s_k|^2)I$, then

$$\mathrm{Re}[F^HF] = c\|H\|^2 I \tag{44}$$

where c is a positive integer. Let

$$\bar{z} = \mathrm{Re}[z], \quad \tilde{z} = \mathrm{Im}[z], \quad \bar{e} = \mathrm{Re}[e], \quad \tilde{e} = \mathrm{Im}[e], \tag{45}$$

and

$$\bar{F}_a = \mathrm{Re}[F_a], \quad \tilde{F}_a = \mathrm{Im}[F_a], \quad \bar{F}_b = \mathrm{Re}[F_b], \quad \tilde{F}_b = \mathrm{Im}[F_b]. \tag{46}$$

Now define

$$z' = \begin{bmatrix} \bar{z} \\ \tilde{z} \end{bmatrix} \quad F' = \begin{bmatrix} \bar{F}_a & \bar{F}_b \\ \tilde{F}_a & \tilde{F}_b \end{bmatrix} \quad e' = \begin{bmatrix} \bar{e} \\ \tilde{e} \end{bmatrix} \tag{47}$$

so that we can write

$$y' = F's' + e' \tag{48}$$

which is actually the same expression as (4) except the vectors and matrices have their rows and columns permuted.

It can be shown that (44) implies

$$F'^TF' = c\|H\|^2 I. \tag{49}$$

Let P_y and P_s be $2K \times 2K$ and $2MT \times 2MT$, respectively, permutation matrices such that

$$\begin{bmatrix} \mathrm{Re}(y_1^1) \\ \mathrm{Im}(y_1^1) \\ \vdots \\ \mathrm{Re}(y_T^M) \\ \mathrm{Im}(y_T^M) \end{bmatrix} = P_y y' \qquad \begin{bmatrix} \mathrm{Re}(s_1) \\ \mathrm{Im}(s_1) \\ \vdots \\ \mathrm{Re}(s_K) \\ \mathrm{Im}(s_K) \end{bmatrix} = P_s s'. \tag{50}$$

It follows that $P_y^TP_y = P_yP_y^T = I$ and $P_s^TP_s = P_sP_s^T = I$.

We now have

$$\begin{aligned} \begin{bmatrix} \mathrm{Re}(y_1^1) \\ \mathrm{Im}(y_1^1) \\ \vdots \\ \mathrm{Re}(y_T^M) \\ \mathrm{Im}(y_T^M) \end{bmatrix} &= P_y(F's' + e') && (51) \\ &= P_yF'P_s^T \begin{bmatrix} \mathrm{Re}(s_1) \\ \mathrm{Im}(s_1) \\ \vdots \\ \mathrm{Re}(s_K) \\ \mathrm{Im}(s_K) \end{bmatrix} + P_ye' && (52) \\ &= \check{H} \begin{bmatrix} \mathrm{Re}(s_1) \\ \mathrm{Im}(s_1) \\ \vdots \\ \mathrm{Re}(s_K) \\ \mathrm{Im}(s_K) \end{bmatrix} + \begin{bmatrix} \mathrm{Re}(v_1^1) \\ \mathrm{Im}(v_1^1) \\ \vdots \\ \mathrm{Re}(v_T^M) \\ \mathrm{Im}(v_T^M) \end{bmatrix}. && (53) \end{aligned}$$

Therefore,

$$\check{H} = P_yF'P_s^T \tag{54}$$

which implies

$$\check{H}^T\check{H} = P_sF'^TP_y^TP_yF'P_s^T = c\|H\|^2I. \tag{55}$$

In other words, $\sigma = c\|H\|^2$. This has an impact on the computational complexity formula (17) which we discuss next.

First, let $c = 1$. Since $\sigma = \|H\|^2$, its calculation takes $2MN$ real multiplications and $2MN - 1$ real additions. As a result, the computational complexity formula (17) can be updated as

$$\mathcal{C}_{\text{PR}} = (4KMT + 2MN + 2K + 4)R_M, (4KMT + 2MN - 2K - 1)R_A. \tag{56}$$

When $c > 1$, the number of real multiplications to calculate σ increases by 1, however, the complexity of the calculation of $\check{H}^T\check{y}$ will reduce by a factor of c, as seen in the examples.

As seen in the examples, the presence of values of 0 within $\check{H}$ will reduce the computational complexity. Its effect will be a reduction in the number of real multiplications to calculate $\check{H}^T\check{y}$ by a factor equal to the ratio of the rows of A_k and B_k that consist only of 0 values to the total number of all rows in A_k and B_k for $k = 1, 2 \ldots, K$, with a similar (not same) reduction in the number of real additions to calculate $\check{H}^T\check{y}$. It will also reduce the number of real multiplications and additions to calculate σ but that effect can be more complicated, as seen in Example 4. Also, as seen in Example 4, the contents of the $\check{H}$ matrix can have linear combinations of h_i values, which also result in changes in computational complexity.

VI. Discussion

For an OSTBC $\mathcal{G}_N$ satisfying $\mathcal{G}_N^H\mathcal{G} = c(\sum_{k=1}^K |s_k\|^2)I$ where c is a positive integer, the Maximum Likelihood solution is formulated in four equivalent ways

$$\|Y - \mathcal{G}_NH\|^2 = \|z - Fs'\|^2 = \|z' - F's'\|^2 = \|\check{y} - \check{H}x\|^2. \tag{57}$$

There are four solutions, all equal. The first solution is obtained by expanding $\|Y - \mathcal{G}_NH\|^2$ and is given by (20) when $c = 1$ [4, eq. (7.4.2)]. When $c > 1$, it should be altered as

$$\hat{s}_k = \frac{1}{c\|H\|^2}[\text{Re}\{\text{Tr}(H^HA_k^HY)\} - \hat{\imath}\cdot\text{Im}\{\text{Tr}(H^HB_k^HY)\}] \qquad k = 1, 2, \ldots, K. \tag{58}$$

The second solution is obtained by expanding the second expression in (57) and is given by

$$\hat{s}' = \frac{\text{Re}[F^Hz]}{c\|H\|^2}. \tag{59}$$

This is given in [4. eq. (7.4.20)] for $c = 1$. The third solution is the solution to the third equation in (57)

$$\hat{s}' = \frac{F'^Tz'}{c\|H\|^2}. \tag{60}$$

The fourth solution is the one introduced in [1]. It is the solution to the fourth equation in (57) and is given by

$$\begin{bmatrix} \text{Re}(\hat{s}_1) \\ \text{Im}(\hat{s}_1) \\ \vdots \\ \text{Re}(\hat{s}_K) \\ \text{Im}(\hat{s}_K) \end{bmatrix} = \frac{\check{H}^T\check{y}}{\sigma} = \frac{\check{H}^T\check{y}}{c\|H\|^2}. \tag{61}$$

Considering that

$$F_a = [\text{vec}(HA_1) \ \cdots \ \text{vec}(HA_K)] \quad F_b = [\hat{\imath}\text{vec}(HB_1) \ \cdots \ \hat{\imath}\text{vec}(HB_K)] \tag{62}$$

[4, eq. (7.1.7)], it can be verified that (58) and (59) are equal. The equality of (59) and (60) follows from (45)-(47). The equality of (60) and (61) follows from (50) and (54). Therefore, equations (58)-(61) yield the same result, and when properly implemented, will have identical computational complexity.

Finally, we would like to state that a straightforward implementation of (58) or (59) can actually result in larger complexity than (60) and (61). The proper implementation requires that in (58) and (59), the terms not needed due to elimination by the Tr[ ], Re[ ], and Im[ ] operators are not calculated. We calculated the computational complexity values for the examples taking this fact into account.

References

[1] L. Azzam and E. Ayanoglu, “A novel maximum likelihood decoding algorithm for orthogonal space-time block codes,” *IEEE Transactions on Communications*, vol. 57, pp. 606–609, March 2009.

[2] ——, “Low-complexity maximum likelihood detection of orthogonal space-time block codes,” in *Proc. IEEE Global Telecommunications Conference*, November 2008.

[3] ——, “Reduced complexity sphere decoding for square QAM via a new lattice representation,” in *Proc. IEEE Global Telecommunications Conference*, November 2007.

[4] E. G. Larsson and P. Stoica, *Space-Time Block Coding for Wireless Communications*. Cambridge University Press, 2003.

[5] V. Tarokh, H. Jafarkhani, and R. Calderbank, “Space-time block coding for wireless communications: Performance results,” *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 451–460, July 1999.